



**BASTA!**  
NET, WINDOWS, VISUAL STUDIO

Thomas Claudius Huber | Trivadis AG

# Neuerungen im Windows 8.1 UI für Entwickler

# Thomas Claudius Huber

- Principal Consultant @ Trivadis AG
  - Trainer, Coach, Developer, Architect
  - [www.thomasclaudiushuber.com](http://www.thomasclaudiushuber.com)
- Spezialisiert auf WPF, XAML, WinApps
- Autor der umfassenden Handbücher zu WPF, Silverlight und Windows Store Apps



# Sessioninhalt

- Neue Controls
- Fensteransichten und Kacheln
- Weitere Neuerungen
  - Binding
  - RenderTargetBitmap
  - ...

# DatePicker

- Date-Property enthält das Datum (DateTimeOffset)
- Formatierung: Day-, Month-, YearFormat
- Weitere Properties: DayVisible, MonthVisible, MaxYear, MinYear, Orientation, ...

```
<DatePicker Header="Datum:"  
  DayFormat="{day.integer} {dayofweek.abbreviated}"  
  DateChanged="OnDateChanged" />
```

Datum:

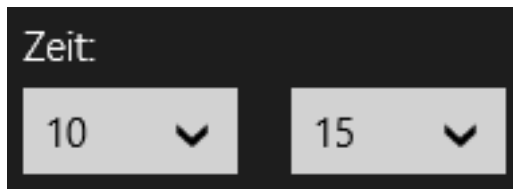
September ▼	24 Tue ▼	2013 ▼
-------------	----------	--------

**Demo**

# TimePicker

- *Time*-Property enthält die Zeit (*TimeSpan*)
- *ClockIdentifier*-Property bestimmt 12/24h-Format
- Weitere Properties: *MinuteIncrement*, *Header*, ...

```
<TimePicker Header="Zeit:" ClockIdentifier="24HourClock"  
    TimeChanged="OnTimeChanged" Margin="10" />
```





# AppBar Features

- *Keine StandardStyles.xaml-Datei mehr*
- Stattdessen neue Klassen namens *AppBarButton*, *AppBarSeparator*, *AppBarToggleButton*
- *CommandBar* mit *Primary* und *SecondaryCommands*

# AppBarButton

- *Button*-Subklasse speziell für die *AppBar*
- Inhalt via *Icon*- und *Label*-Property
- *IsCompact*-Property entfernt Margin und Label

```
<!--Kurze Schreibweise-->  
<AppBarButton Icon="BrowsePhotos" Label="Bilder"/>
```

```
<!--Lange Schreibweise-->  
<AppBarButton Label="Bilder">  
    <AppBarButton.Icon>  
        <SymbolIcon Symbol="BrowsePhotos"/>  
    </AppBarButton.Icon>  
</AppBarButton>
```

**Demo**

# AppBarButton-Icons

- SymbolIcon
- FontIcon
- PathIcon
- BitmapIcon

```
<AppBarButton Label="Turbo">  
  <AppBarButton.Icon>  
    <BitmapIcon UriSource="ms-appx:///turbo.png"/>  
  </AppBarButton.Icon>  
</AppBarButton>
```

**Demo**



# CommandBar

- Direkte Subklasse von AppBar
- Definiert die beiden Properties PrimaryCommands und SecondaryCommands
- Ordnet die Commands gemäss UX-Richtlinie an

```
<CommandBar>  
  <CommandBar.PrimaryCommands>  
    <!--Primäre Commands auf der rechten Seite-->  
  </CommandBar.PrimaryCommands>  
  <CommandBar.SecondaryCommands>  
    <!--Sekundäre Commands auf der linken Seite-->  
  </CommandBar.SecondaryCommands>  
</CommandBar>
```

**Demo**

# Flyout and MenuFlyout

- Werden der *Flyout*-Property eines *Button* zugewiesen
  - via Attached-Property *FlyoutBase.AttachedFlyout* jedem beliebigen *FrameworkElement*
- *Flyout*: zeigt ein beliebiges *UIElement* an
- *MenuFlyout*: zeigt ein Menu an:
  - Kindelemente sind *MenuFlyoutItem*, *ToggleMenuFlyoutItem* und *MenuFlyoutSeparator*

**Demo**

# SettingsFlyout

- Neues Control zur Anzeige der Einstellungen
  - Beachtet die UI-Richtlinien
- Show-Methode -> zurück zu den Settings
- ShowIndependent-Methode -> zurück zur App



← BastaSettingsFl... ✕

BASTA! 2013

Yesss, it's BASTA!-Time

Automatischer NSA-Sessionupload  
aktivieren

# Das Hub-Element

- Stellt beliebige Inhalte in Panorama-Ansicht dar
- Besteht aus einzelnen HubSections
- VS2013 besitzt neue «Hub App»-Vorlage

```
<Hub Header="BASTA!" SectionHeaderClick="OnHeaderClick">  
  <HubSection Header="Speaker" IsHeaderInteractive="True">  
    <DataTemplate>...</DataTemplate>  
  </HubSection>  
  <HubSection Header="Sessions">  
    <DataTemplate>...</DataTemplate>  
  </HubSection>  
  ...  
</Hub>
```



**Demo**

# Sonstiges

- Weitere Verbesserungen existierender Controls
  - Header-Property für TextBox, ComboBox etc.
  - PlaceholderText-Property für TextBox, ComboBox etc.
  - ...
- Hyperlink-Element für TextBlock/RichTextBlock
- SearchBox, um Suche in App zu integrieren

**Demo**

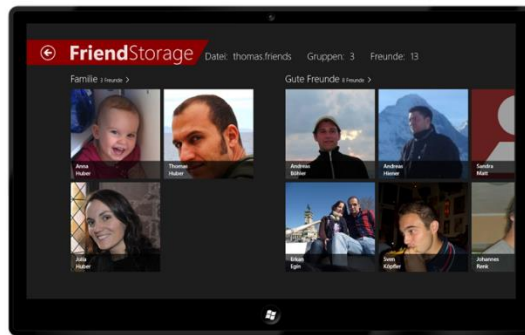


# Sessioninhalt

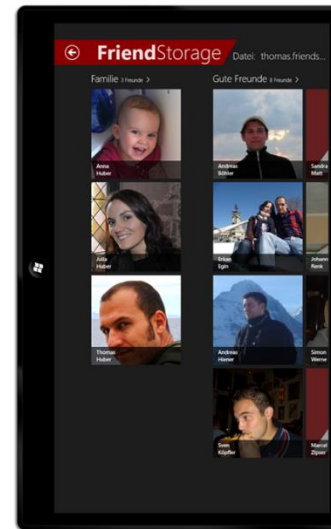
- Neue Controls
- Fensteransichten und Kacheln
- Weitere Neuerungen
  - Binding
  - RenderTargetBitmap
  - ...

# Fensteransichten in Win 8

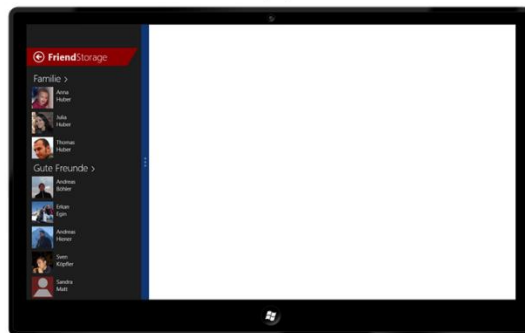
## FullScreenLandscape



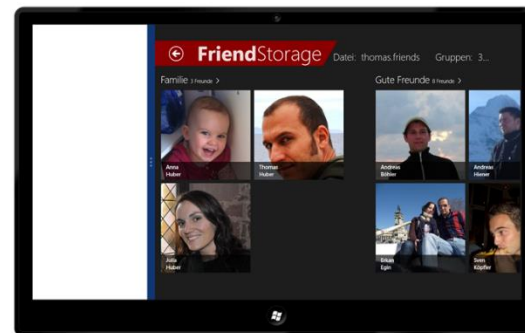
## FullScreenPortrait



## Snapped



## Filled



# Fensteransichten in Win 8.1

- Keine feste Breite mehr
  - Benutzer können Apps bis zur Mindestbreite beliebig breit darstellen
- Mindestbreite ist standardmäßig 500px
  - Snapped-View in Windows 8 war 320px
- Mindestbreite kann auf 320px verringert werden
  - Via Package.appxmanifest

**Demo**

# Obsolete Properties/Enums

- `ApplicationView.Value` – Property obsolete, da es keine Ansichtszustände mehr gibt
- `ApplicationViewState` – Aufzählung obsolete, da es keine Ansichtszustände mehr gibt
- *`ApplicationView.TryUnsnap`* – nicht mehr gültig, da Apps keine Snapped-Ansicht mehr besitzen

# Neue Properties der ApplicationView-Klasse

- `AdjacentToLeftDisplayEdge/`  
`AdjacentToRightDisplayEdge`
- `IsFullScreen`
- `Orientation` – Landscape oder Portrait



# Mehrere App-Fenster

- *CoreApplication.CreateNewView*-Methode zum Erstellen einer neuen View
- *ApplicationViewSwitcher*.  
*TryShowAsStandaloneAsync* zum Anzeigen
  - Optional mit *ProjectionManager* auf zweitem Bildschirm die neue View darstellen
- Windows 8.1 Samples: Multiple Views

**Demo**

# Neue Kachel-Größen

- Windows 8 hatte 2 Kachel-Größen:
  - Square (150x150)
  - Wide (310x150)
- Windows 8.1 führt zwei weitere Größen ein:
  - Small (70x70)
  - Large (310x310)
- In Windows 8.1 wird die «Square»-Größe als «Middle» bezeichnet
  - Es gibt Small, Middle, Large und Wide

# Neue Kachel-Größen



# Live-Tiles

- TileTemplateType-Enum wurde erweitert
- Mitglieder haben jetzt die Skalierung enthalten:
  - TileSquare150x150\*
  - TileSquare310x310\*
  - TileWide310x150\*
- Alte Mitglieder ohne Skalierung nicht mehr verwenden:
  - TileSquareImage
  - TileWideImage
  - ...

**Demo**

# Sessioninhalt

- Neue Controls
- Fensteransichten und Kacheln
- Weitere Neuerungen
  - Binding
  - RenderTargetBitmap
  - ...



# Binding-Neuerungen

- UpdateSourceTrigger mit PropertyChanged
- Zugriff auf die BindingExpression
- TargetNullValue und FallbackValue

# RenderTargetBitmap

- Subklasse von ImageSource
- RenderAsync-Methode zeichnet UIElement
- GetPixelsAsync um Bild zu speichern/teilen

**Demo**

# Themes

- RequestedTheme-Property auf jedem Element
  - In Windows 8 nur auf App-Objekt
- ThemeResource-Markup Extension erlaubt dynamische Änderung des Themes
- IntelliSense hilft, die Ressourcen zu finden.

# Sonstiges

- Performance-Optimierungen
  - WebView, MediaPlayer
  - Zahlreiche neue WinRT-APIS
- 
- Siehe Windows 8.1 Feature Guide unter <http://dev.windows.com>:  
<http://msdn.microsoft.com/library/windows/apps/bg182410>

# Danke

Twitter: @thomasclaudiush

Homepage: [www.thomasclaudiushuber.com](http://www.thomasclaudiushuber.com)

Mail: [thomas.huber@trivadis.com](mailto:thomas.huber@trivadis.com)

Slides/Demos:

[www.thomasclaudiushuber.com/blog](http://www.thomasclaudiushuber.com/blog)